

### REMARKS

Reconsideration of this application is respectfully requested in view of the foregoing amendment and the following remarks.

Claims 1-26 were pending in this application. Claims 12 and 19 have been cancelled, claims 27 and 28 have been added and claims 1, 10, 11 and 14 have been amended hereby. Support for the Amendment to independent claims 1 and 14 can be found in now-cancelled claims 12 and 19 and paragraphs 0035 of the specification. Claims 10 and 11 have been amended to correct a minor matter of form. Support for new claims 27 and 28 can be found, for example, in paragraphs 0040 and 0041 of the present specification. For the reasons stated below, Applicants respectfully submit that all claims pending in this application are in condition for allowance.

In the Office Action, claims 10, 11 and 17 were rejected under 35 U.S.C. §112, second paragraph. It is believed that the amendments made to claims 10, 11 and 14 address the concerns raised in the Office Action. Accordingly, Applicants respectfully request that this ground of rejection be withdrawn.

Also in the Office Action, claims 1-5 were rejected under 35 U.S.C. §102(b) as being anticipated by Subrahmanian et al., U.S. Patent No. 5,671,352, and claims 6-21 were rejected under 35 U.S.C. §103(a) as being unpatentable over Subrahmaniam et al. in view of Tierney et al., U.S. Patent No. 5,513,315. To the extent these grounds of rejection might still be applied to claims presently pending in this application, they are respectfully traversed.

The present invention is directed to methods for assessing how long continuously operating software systems can be expected to remain executing in a safe and/or reliable manner

before anomalous conditions will ultimately lead to failure. In accordance with the present invention, two separate actions are taken with respect to the continuously operating software systems. First, while the software system is operating, an anomaly is injected into the software system and the output of the system is monitored for unacceptable output. Second, an assertion is employed to help analyze the effect of the injected anomaly and, specifically, to trap values known to be hazardous. More particularly, as recited in amended claim 1, a first data state anomaly is injected in the software system. An assertion is also inserted in the software system. The software system is run after the data state anomaly has been injected and, an unacceptable output from the software is monitored. If such an unacceptable output is observed, it is logged. The assertion, in the meanwhile, is used to trap values known to produce hazardous outputs.

Similarly, amended claim 14 recites initializing the software system and inserting an assertion into the software system. The software system is then run for a first predetermined period after which it is paused and a first data state anomaly is injected. The software system is then run again and the output from the software system is checked. If an unacceptable output is observed, the software system is stopped, and if no unacceptable behavior is observed, the software system is stopped after a second predetermined period has elapsed. Meanwhile, the assertion operates to trap values known to produce hazardous outputs.

In addition to the foregoing, new claims 27 and 28 recite a specific methodology for setting a safe operating period for a continuously operating software system. In accordance with these claims (and as explained in, e.g., paragraphs 0040 and 0041) a safe operating period is set by adding together a time before insertion of an anomaly and the time after insertion of the anomaly during which no unacceptable behavior has been observed. In this manner, there can be

confidence in the operating system that a catastrophic failure will not occur since testing has shown that unacceptable output has not been observed for at least these two successive periods.

The prior art of record fails to disclose the features of the present invention discussed above. Subrahmaniam et al. describe error injection to a behavioral model. In the specific example given, at col. 3, lines 21-52, it is explained how to test whether or not a simulated system is capable of overcoming misalignment errors, which typically occur in memory management. In this case, an inject error command may be directed to memory management 208's UIC 222. UIC 222, in turn, logs the command to test whether error handling code is capable of correctly overcoming the misalignment error occurring in memory management 208.

Once the error injection command is logged in the user interface module UIC 222 of the simulated module to be tested, simulated system 202 continues its execution. Once the instructions reach a certain address range as specified in the error injection command supplied by the user, the type of error to be tested is specified by the user is caused in the specific simulated module to be tested. When the error is caused, the simulated module takes the same action as would be done by the actual hardware which may result in a trap. If the trap is enabled, the error handler is tested. Simulated system 202 continues execution until it hits another error address range within which a certain specified error needs to be tested.

From the foregoing, it is clear that while Subrahmaniam et al. discloses the practice of injecting errors into specific points in a software system, the methodology described does not include inserting "assertions" that are used to see how the injected error affects the software system, as recited in the independent claims. Subrahmaniam et al. only describe how to confirm whether an error handler is operating properly.

Moreover, neither Subrahmaniam et al., nor any other prior art of record, disclose or suggest the features of new claims 27 and 28, which specifically recite how to set a safe operating period for a continuously operating software system by summing together two periods during which unacceptable output has not been observed.

Since the prior art of record fails to disclose or to suggest each and every feature recited in the claims, Applicants respectfully request the §102 and §103 rejection of the claims be reconsidered and withdrawn.

In view of the foregoing all of the claims in this case are believed to be in condition for allowance. Should the Examiner have any questions or determine that any further action is desirable to place this application in even better condition for issue, the Examiner is encouraged to telephone applicants' undersigned representative at the number listed below.

SHAW PITTMAN LLP  
1650 Tysons Boulevard  
McLean, VA 22102  
Tel: 703/770-7900

Date: July 20, 2004

Respectfully submitted,

JEFFREY M. VOAS ET AL.

By:



Lawrence D. Eisen  
Registration No. 41,009

Attachments: None

LDE/dkp

Customer No. 28970